Notes on Influence Maximization

Bhargav Samineni

Abstract

A brief collection of notes that summarizes the Influence Maximization problem in networks and techniques and algorithms used to create approximate solutions.

1 Introduction

In social networks, a natural question to consider is how ideas and information may spread through them. Motivated by applications to viral marketing, Domingos and Richardson [1] considered this problem from the perspective of finding an initial set of nodes that could be targeted (for example with free samples) such that these nodes would spread the influence of a product throughout the network. Influence would be spread through a "word of mouth" effect where the initial nodes would recommend product adoption to their connections, who recommend it to their connections, and so on. This would cause a cascade of adoption in the network, with the ultimate goal of achieving the largest expected number of influenced nodes at the end of the process. This type of problem formulation has applications beyond marketing, such as in the study of how infectious diseases spread.

1.1 Influence Maximization

Kempe et al. [3, 2] later formalized this model as a discrete optimization problem concerned with the spread of influence in a network. Given some stochastic model of diffusion through a network and a node set A, the number of active (i.e. influenced) nodes at the end of a diffusion cascade started by A is denoted by $\varphi(A)$. Note that this is a random variable as the diffusion model is probabilistic. The *influence* of A, denoted by $\sigma(A)$, is then the expected value of the number of active nodes at the end of the diffusion process started by A (i.e. $\sigma(A) = \mathbb{E}[\varphi(A)]$). The Influence Maximization (InfMax) problem looks to find, for a parameter k, a "seed" set of nodes of size k that maximizes influence. Under most diffusion models, this problem is NP-Hard [3, 2].

Problem 1 (InfMax)

Given a digraph $G = (V, E)$, a model for diffusion, and an integer k, find a seed set $S \subseteq V$ of size k such that $\sigma(S)$ is maximized.

1.2 Diffusion Models

For the purposes of a diffusion model, nodes in the network can either be active or inactive. These models are typically progressive, meaning that once a node becomes active, it cannot turn inactive. Denote the set of in-neighbors of a node v by $\Gamma(v)$. Two common types of diffusion models are described below.

Definition 1 (LT Model)

The *Linear Threshold* model is defined by a digraph $G = (V, E)$ and an edge weight function $w\colon E \to (0, 1]$, where each node $v \in V$ is associated with some threshold θ_v and the weight of an edge $(u, v) \in E$ denotes the degree of influence u has on v. Notably, θ_v is sampled uniformly at random from $[0, 1]$ and $\sum_{\{u \mid u \in \Gamma(v)\}} w(u, v) \leq 1$ for each $v \in V$. On an input set S, the model works as follows:

1. In time step 0, the nodes in S become activated
2. In time step t, a node v becomes activated if

$$\sum_{\{u \mid u \in \Gamma(v) \text{ and } u \text{ is active}\}} w(u, v) \geq \theta_v$$

3. The process ends after there is a time step with no additional activated nodes

This process takes at most $|V|$ time steps.

Definition 2 (IC Model)

The *Independent Cascade* model is defined by a digraph $G = (V, E)$ and an edge weight function $p\colon E \to (0, 1]$, where the weight of each edge $(u, v) \in E$ represents the probability that u activates v. On an input set S, the model works as follows:

1. In time step 0, the nodes in S become activated
2. In time step t, each newly activated node u in time step $t - 1$ gets one chance to activate each inactivated out-neighbor v with probability $p(u, v)$
3. The process ends after there is a time step with no additional activated nodes

This process takes at most D time steps, where D is the diameter of G.

2 Approximation Algorithms

Kempe et al. [3] proved that under both the LT and IC models, an optimal solution to InfMax can be approximated to within a factor of $\left(1 - \frac{1}{e} - \varepsilon\right)$, where $\varepsilon \in [0, 1 - \frac{1}{e}]$. The core of their argument is that under these models, the influence function σ satisfies the property of submodularity.

Definition 3 (Submodularity)

Let U be some universal set. A function $f\colon U \to \mathbb{R}$ is submodular if it satisfies

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \tag{1}$$

for all sets $A \subseteq B \subseteq U$ and elements $e \in U \setminus B$.

If it can be shown that σ is non-negative, monotone (i.e. non-decreasing), and submodular for some diffusion model, then a result by Nemhauser et al. [4] shows that a simple greedy algorithm

for maximizing such a submodular function subject to a cardinality constraint gives a $\left(1 - \frac{1}{e}\right)$ approximation to the optimal solution [1].

Theorem 1 ([4]). *Consider a non-negative, monotone, submodular function f and an integer k. Let S^* be the set of size k that maximizes f (i.e. the optimal solution). Let S be the set constructed by iteratively choosing the k elements that give the maximum marginal gain to the function value at each iteration, starting from the empty set. Then $f(S) \geq \left(1 - \frac{1}{e}\right) f(S^*)$.*

A generalization of the above theorem, where the greedy algorithm iteratively picks elements within a factor of $(1 - \varepsilon)$ of the largest marginal gain at each step gives a $(1 - \frac{1}{e} - \varepsilon')$ approximation, where ε' depends polynomially on ε.

2.1 Submodularity under the Independent Cascade Model

Because of the probabilistic nature of σ, it is difficult to prove it is submodular directly. Instead, it is helpful to reformulate the problem in terms of *realizations* of the underlying network.

Definition 4 (Realization)

A realization of a graph $G = (V, E)$ can be thought of as a mapping $\phi \colon E \to \{L, B\}$, where edges are given a label L if they are a *live* edge and B if they are a *blocked* edge. A node can be considered active iff there is a path consisting of only live edges between it and another active node.

This defines a probability space of 2^m realizations, where each realization denotes a unique configuration of live and blocked edges. In the IC model, an edge (u, v) is live with probability $p(u, v)$ and blocked with probability $1 - p(u, v)$.

Theorem 2. *For any instance of the IC model, the influence function σ is submodular.*

Proof. Consider some arbitrary realization ϕ of the network G. We can now define the function $\varphi_\phi(S)$ that gives the number of active nodes in the realization ϕ by initially activating a seed set S. Note that this is in fact now a deterministic quantity because we have fixed our choice of ϕ. Hence, $\sigma_\phi(S) = \mathbb{E}\left[\varphi_\phi(S)\right] = \varphi_\phi(S)$ is also deterministic.

We now show that σ_ϕ is submodular. Define $R(v, \phi)$ to be the set of nodes that have a direct path from node v consisting of only live edges. Let A, B be two seed node sets such that $A \subseteq B$ and v a node not in B. With A, the addition of v to it increases $\sigma_\phi(A)$ by $|R(v, \phi) \setminus \cup_{u \in A} R(u, \phi)|$, and similarly with B it increases $\sigma_\phi(B)$ by $|R(v, \phi) \setminus \cup_{u \in B} R(u, \phi)|$. However, since clearly $\cup_{u \in A} R(u, \phi) \subseteq \cup_{u \in B} R(u, \phi)$, the amount added to $\sigma_\phi(A)$ is at least the amount added to $\sigma_\phi(B)$. Hence,

$$\sigma_\phi(A \cup \{v\}) - \sigma_\phi(A) \geq \sigma_\phi(B \cup \{v\}) - \sigma_\phi(B)$$

which satisfies Def. 3.

Note that

$$\sigma(S) = \sum_{\text{realizations } \phi} \Pr(\phi)\, \sigma_\phi(S)$$

[1] [5] gives a greedy algorithm with the same approximation ratio for the more general problem of maximizing a non-negative, monotone, submodular function under a knapsack constraint.

since the expected number of active nodes at the end of a cascade starting from S is just the weighted average of the number of active nodes found by activating S over all realizations. Since a non-negative linear combination of submodular functions is also submodular, σ is submodular. ■

2.2 Submodularity under the Linear Threshold Model

We again make use of the concept of realizations in Def. 4. However, in general, fixing a choice of threshold values does not guarantee that the influence function will be submodular. Thus, we require a different analysis compared to the IC model.

In the LT model, an edge cannot be considered live or blocked independently of the status of other edges. Instead, at most one incoming edge for each node can be considered live. More formally, a node v selects at most one its in-edges at random, where each edge is chosen with probability $w(u, v)$ where $u \in \Gamma(v)$, or chooses no edge with probability $1 - \sum_{u \in \Gamma(v)} w(u, v)$ to be live. The rest of the edges are considered blocked.

Claim 1. *For a seed set S, the following result in the same distributions over active sets of nodes:*

1. *Running the Linear Threshold process to completion starting from S*
2. *Creating realizations as defined above and initially activating S.*

Proof. We first consider the LT process. We can extend the notion of influence weight by defining $w(u, v) = 0$ if $u \notin \Gamma(v)$. Define S_t to be the set of active nodes at the end of iteration t, where $t = 0, 1, \ldots$ and $S_0 = S$. If a node v has not become active at the end of iteration t, the probability it becomes active at the end of iteration $t + 1$ is equal to the chance that the influence weights in $S_t \setminus S_{t-1}$ push it over its threshold given that its threshold has not already been passed. From a notational standpoint, this is equivalent to

$$\Pr\left(\theta_v \leq \sum_{u \in S_t} w(u, v) \;\middle|\; \theta_v > \sum_{u \in S_{t-1}} w(u, v)\right) = \frac{\Pr\left(\sum_{u \in S_{t-1}} w(u, v) < \theta_v \leq \sum_{u \in S_t} w(u, v)\right)}{\Pr\left(\theta_v > \sum_{u \in S_{t-1}} w(u, v)\right)} = \frac{\sum_{u \in S_t \setminus S_{t-1}} w(u, v)}{1 - \sum_{u \in S_{t-1}} w(u, v)}$$

since θ_v is sampled uniformly at random from $[0, 1]$.

We now consider the realization process, gradually revealing active nodes in iterations as in the LT model. Define S_t for iterations $t = 0, 1, \ldots$ as before and additionally define $S'_t = S_t \setminus S_{t-1}$ where $S'_0 = S_0$. Starting with S'_0, for each node v with at least one in-edge from S'_0, determine if v has a live edge coming from this set. If so, then v becomes active with $v \in S'_1$; otherwise we keep the source of v's live in-edge (if it has one) unknown. Repeat this process starting from S'_1, thus defining sets $S'_2, S'_3, \ldots$. If v has not become active at the end of iteration t, then the probability it is determined to be active at the end of iteration $t + 1$ is equivalent to the chance that its live edge comes from S'_t given that its live edge has not come from $S'_0, \ldots, S'_{t-1}$. Namely, this probability is

$$\frac{\sum_{u \in S'_t} w(u, v)}{1 - \sum_{u \in \cup_{i=0}^{t-1} S'_i} w(u, v)} = \frac{\sum_{u \in S_t \setminus S_{t-1}} w(u, v)}{1 - \sum_{u \in S_{t-1}} w(u, v)}.$$

Since these probabilities are equivalent in both the LT model and realization process, we have the proof of the claim. ■

Since we have established that the notion of activations through realizations and live edge paths as defined before is equivalent to the LT model, we get the following theorem whose proof is identical to that of Theorem 2.

Theorem 3. *For any instance of the LT model, the influence function σ is submodular.*

A Generalization of Theorem 1

Theorem 4. *Consider a non-negative, monotone, submodular function f and an integer k. Let S^* be the set of size k that maximizes f (i.e. the optimal solution). Choose some $0 \leq \varepsilon \leq \frac{1}{k}$. If S is the set constructed by iteratively choosing k elements that maximize the marginal gain of the function value to within a factor of $(1-\varepsilon)$ at each step, starting from the empty set, then $f(S) \geq \left(1 - \frac{1}{e} - \varepsilon\right) f(S^*)$.*

Let $S^* = \{s_1^*, \ldots, s_k^*\}$ be the optimal solution and $S = \{s_1, \ldots, s_k\}$ the solution constructed by the above greedy algorithm. Define $S_i^* = \{s_i^*, \ldots, s_i^*\}$ and similarly $S_i = \{s_1, \ldots, s_i\}$ for $i = 0, \ldots, k$, where $S_0^* = \emptyset = S_0$. We first prove the following two claims.

Claim 2.

$$f(S_{i+1}) - f(S_i) \geq \frac{1-\varepsilon}{k} \left(f(S_i \cup S^*) - f(S_i)\right) \qquad \text{for each } i = 0, \ldots, k-1.$$

Proof. Choose some $0 \leq i \leq k-1$. By definition of the greedy approach,

$$\begin{aligned} f(S_{i+1}) - f(S_i) &\geq (1-\varepsilon) \left(\max_{s \in U} \left(f(S_i \cup \{s\}) - f(S_i)\right) \right) \\ &\geq (1-\varepsilon) \frac{1}{k} \sum_{j=1}^{k} \left(f\left(S_i \cup \left\{s_j^*\right\}\right) - f(S_i)\right) \end{aligned} \tag{2}$$

where the second inequality comes from the fact that since s is a maximizer, an average of the marginal gain from others elements in U (in this case specifically in S^*) must be at most the marginal value s provides.

By submodularity of f,

$$\begin{aligned} f\left(S_i \cup \left\{s_j^*\right\}\right) - f(S_i) &\geq f\left(S_i \cup S_{j-1}^* \cup \left\{s_j^*\right\}\right) - f\left(S_i \cup S_{j-1}^*\right) \\ &= f\left(S_i \cup S_j^*\right) - f\left(S_i \cup S_{j-1}^*\right) \end{aligned}$$

since $S_i \subseteq S_i \cup S_{j-1}^*$. Using this inequality with Eq. (2), we get

$$\begin{aligned} f(S_{i+1}) - f(S_i) &\geq \frac{1-\varepsilon}{k} \sum_{j=1}^{k} \left(f\left(S_i \cup S_j^*\right) - f\left(S_i \cup S_{j-1}^*\right)\right) \\ &= \frac{1-\varepsilon}{k} \left(f(S_i \cup S_k^*) - f(S_i \cup S_0^*)\right) \\ &= \frac{1-\varepsilon}{k} \left(f(S_i \cup S^*) - f(S_i)\right) \end{aligned}$$

where the equality comes from the fact that the summation telescopes. ∎

Claim 3.

$$f(S_i) \geq \left(1 - \left(1 - \frac{1}{k}\right)^i - \varepsilon\right) f(S^*) \qquad \text{for each } i = 1, \ldots, k.$$

Proof. To simplify analysis, we assume WLOG that $f(\emptyset) = 0$. We prove by induction. The base case $i = 1$ follows from Claim 2:

$$\begin{aligned} f(S_1) = f(S_1) - f(S_0) &\geq \frac{1-\varepsilon}{k} \left(f(S_0 \cup S^*) - f(S_0)\right) \\ &> \left(\frac{1}{k} - \varepsilon\right) f(S^*) \,. \end{aligned}$$

For the induction case, assume the claim holds for $i = j$. Then,

$$\begin{aligned} f(S_{j+1}) - f(S_j) &\geq \frac{1-\varepsilon}{k} \left(f(S_j \cup S^*) - f(S_j)\right) && \text{(by Claim 2)} \\ &\geq \frac{1-\varepsilon}{k} \left(f(S^*) - f(S_j)\right) && \text{(by monotonicity)} \\ f(S_{j+1}) &\geq \frac{1-\varepsilon}{k} f(S^*) + \frac{k-1+\varepsilon}{k} f(S_j) \\ &\geq \frac{1-\varepsilon}{k} f(S^*) + \frac{k-1+\varepsilon}{k} \left(1 - \left(1 - \frac{1}{k}\right)^j - \varepsilon\right) f(S^*) && \text{(by ind. hyp.)} \\ &= f(S^*) \left(\frac{1}{k} - \frac{\varepsilon}{k} + \left(1 - \frac{1}{k} + \frac{\varepsilon}{k}\right) \left(1 - \left(1 - \frac{1}{k}\right)^j - \varepsilon\right)\right) \\ &= f(S^*) \left(1 - \left(1 - \frac{1}{k}\right)^{j+1} - \varepsilon + \frac{\varepsilon}{k} \left(1 - \left(1 - \frac{1}{k}\right)^j - \varepsilon\right)\right) \\ &\geq f(S^*) \left(1 - \left(1 - \frac{1}{k}\right)^{j+1} - \varepsilon + \frac{\varepsilon}{k} \left(\frac{1}{k} - \varepsilon\right)\right) && (j = 1 \text{ is the maximizer}) \\ &= f(S^*) \left(1 - \left(1 - \frac{1}{k}\right)^{j+1} - \left(\frac{\varepsilon k^2 - \varepsilon + \varepsilon^2 k}{k^2}\right)\right) \\ &\geq f(S^*) \left(1 - \left(1 - \frac{1}{k}\right)^{j+1} - \varepsilon\right), \end{aligned}$$

which proves the induction case. ■

Proof of Theorem 4. Take $i = k$ in Claim 3. Then

$$f(S_k) = f(S) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k - \varepsilon\right) f(S^*) > \left(1 - \frac{1}{e} - \varepsilon\right) f(S^*) \,.$$

■

References

[1] P. Domingos and M. Richardson. "Mining the Network Value of Customers". In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2001, pp. 57–66.

[2] D. Kempe, J. Kleinberg, and É. Tardos. "Influential Nodes in a Diffusion Model for Social Networks". In: *Proceedings of the 32nd International Conference on Automata, Languages and Programming*. 2005, pp. 1127–1138.

[3] D. Kempe, J. Kleinberg, and É. Tardos. "Maximizing the Spread of Influence through a Social Network". In: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2003, pp. 137–146.

[4] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. "An analysis of approximations for maximizing submodular set functions—I". In: *Mathematical Programming* 14.1 (1978), pp. 265–294.

[5] M. Sviridenko. "A note on maximizing a submodular set function subject to a knapsack constraint". In: *Operations Research Letters* 32.1 (2004), pp. 41–43.